Scheduling & Resource Management Grid Forum Working Group

sched-wg@gridforum.org

HPDC-8 Grid Forum BOF and
Impromptu Working Group Meeting

August 4 & 6, 1999

Bill Nitzberg & Jennifer Schopf
(interim chairs)

Participants [Jun '99]

Bill Nitzberg, NASA Ames / MRJ, bnitzberg@arc.nasa.gov
Mike Peterson, U. Florida, peterson@chem.ufl.edu
Jonathan Geisler, Northwestern U., geisler@ece.nwu.edu
Jon Weissman, U. Texas/U. Minnesota, jon@cs.utsa.edu
Jennifer Schopf, Northwestern U., jms@cs.nwu.edu
David Bader, U. New Mexico, dbader@eece.unm.edu
Greg Hommes, Lawrence Livermore Lab, hommes1@llnl.gov
Judy Beiriger, Sandia National Laboratories, jibeiri@sandia.gov
Barny Maccabe, U. New Mexico, maccabe@cs.unm.edu
Mitch Murphy, MHPCC, mitch@mhpcc.edu
Ed Hook, NASA Ames / MRJ, hook@nas.nasa.gov
Tom Cheatham, Harvard, cheatham@deas.harvard.edu
Dan Stefanescu, Harvard, dan@deas.harvad.edu
James P. Jones, NASA Ames / MRJ, jjones@nas.nasa.gov
Andy Yoo, LLNL, yoo2@llnl.gov
Cas Lesiak, NASA Ames / MRJ, clesiak@nas.nasa.gov
Bhroam Mann, NASA Ames / MRJ, bmann@nas.nasa.gov
Abdul Waheed, NASA Ames / MRJ, waheed@nas.nasa.gov
Mark Clement, BYU, clement@cs.byu.edu
Quinn Snell, BYU, snell@cs.byu.edu
Keith Jackson, LBNL, krjacks@lbl.gov
Gary Hoo, LBNL, gjhoo@lbl.gov
Hyo Jung Song, UCSD, hjsong@csag.ucsd.edu

Potential Areas

- General architecture
- Advance Reservations
- Super-scheduling
- Resource specification
- Events (for notification)
- “Cost model”
- Co-scheduling
- Quality of service
- Quality of information
- Policy and guarantees
- Monitoring
- Accounting, allocations, logging, error handling
- Resource fungability
- Scheduling disk, network, tape, etc.

Charter & Goals

Charter: “Solve grid resource management”

Active areas:

- Advance reservations
- Super-scheduling
- Resource specification & semantics

Goals

- Better definition of charter
- Progress in three areas identified
- Work via mailing list: sched-wg@gridforum.org

Area: Advance Reservations

- Advance reservations
  - Capability: “reserve resources {R} for time period T”
  - Bill Nitzberg, chair
- Goals
  - Prototype reservation across different resource management packages (and sites) [SC 1999]
  - Specification for an API for advance reservations [Jun 2000]

Area: Super-scheduling

- Super-scheduling & “global queueing”
  - Capability: “given a job, run it on Grid resources”
  - Jenny Schopf, chair
- Goals
  - Prototype super-scheduler [SC 1999]

Area: Resource Specification & Semantics

- List of attributes/tokens (resource specification and semantics)
  - Language + tokens
  - Quinn Snell, chair
- Goals
  - list of attribute/value pairs [Oct 1999]
  - Specification for a common intermediate form for job description and resource specification [Jun 2000]

Working Group Meeting: Friday, Noon @ “Capt’n Kidds”

- Directions: 1/2 block left across street

- Introductions
- Break up into groups to discuss active areas:
  - Advance reservations
  - Super-scheduling
  - Resource specification & semantics
  - “Other stuff…”

Advance Reservations Working Group Notes

August 6, 1999

Mark Clement, scribe

Two Types of Reservations

- Time based reservations
  - You want to allocate 3 hours of time and may run different applications during that time. For example, you may be debugging your code and need to go through a few cycles, or you may have some other use that requires multiple applications to be run
- Job based reservations
  - You want to release the resources as soon as your job has completed
- We need to have a way of specifying the reservation type in the submission language. Local policy may dictate this as well.

Callbacks

- When the reservation time arrives, the scheduler should call the metascheduler so it can launch the job. This functionality is difficult to implement in some of the schedulers.

Reservation Delegation

- We need to have a token returned that identifies the reservation and this token should be able to be passed to someone else (delegation of tokens)

Soft or Hard Reservations

- Soft reservations would make a large soft reservation on all of the sub-schedulers. The metasheduler would then decide on the overlapping regions, pick the most desirable time and make a hard reservation. If the hard reservation was not made in a specified period of time, the soft reservation would be automatically released. One disadvantage of this method is that large time regions are locked down during the three phase commit. From the scheduler's perspective, soft reservations are no less expensive than hard reservations. The scheduler must still avoid reserved time regions and will not let other metaschedulers arbitrate for a schedule there.
- With hard reservations, the metascheduler first queries the schedulers to get their current schedules and then starts making hard reservations in the areas that look most desirable. Something may have changed between the query and the hard reservation and so one of the reservations may fail. In this case, all of the reservations are released and the process starts again.
- We have found that hard reservations make a lot more sense. The only time that they are inferior is when the schedule is changing rapidly. Our experience with the Maui scheduler indicates that the schedule only changes about once every 15 minutes. In this case, Hard reservations are preferred. We are currently looking at trace data from many supercomputing centers to determine which method works best at an average center.

Cost / Price

- We talked a lot about the price for making and canceling reservations. Although they get some favorable treatment, reservations will not be backfilled, so they may end up with a fair treatment when compared with non-reserved jobs.

Legion & Reservations

- We asked what Legion would like to see in a reservation system
  - The ability to query the number of processors and the load on each processor
  - The ability to delegate reservations using secure tickets

Super-scheduling Working Group Notes

August 6, 1999

Jenny Schopf, scribe

Super-scheduling

- [Notes to be added.]

Resource Specification & Semantics Working Group Notes

August 6, 1999

Quinn Snell, scribe

Summary

- Because many of the group had not been involved with the Scheduling subgroup before, there was not initially any agreement on whether the grid should specify a language. There was a lot of talk about just letting everybody do what they want so as to not stifle creativity.
- After much talk, the group could see that we should allow everyone to do what they want to do locally, but ask all those that participate in the Grid to translate their language to the grid language. It is much the same as asking everybody who creates a DC electrical appliance to have their own adapter to convert the standard AC power to their required power input.
- There was very little discussed about specifics of the language, except that the language should definitely be declarative.

Notes

- Language issues:
- The first question to answer is if we really need a language specification. An open environment could be created that would use conversion scripts to translate between different job control languages. Current schedulers all speak their own language and require control applications to speak their own dialect for job control.
- XML?
- Several XML like languages have been proposed
- In any scheme you need advertisement of the language. Another alternative is to use the LDAP fixed schema
- We need to recommend what to advertise
- How do we exist in a Multilanguage environment?
- There can be no Exclusion
- With the power grid, it is acceptable to force the user to plug in using an adapter. Something like this may be a good paradigm for the computational power grid.

Notes, cont.

- The language must be adaptable in order to accommodate change, new semantics, simple, flexible
- Look at Condor pool configuration
- Look at Classified Ads
- We need to decide on the level of expression
- Should we go with a declarative or procedural paradigm?
- The group thought it should be Declarative
- What are the minimal units that can be specified?
- What are the structures you can create?

“Other Stuff…”
Working Group Notes

August 6, 1999

Bill Nitzberg, scribe

Want (from queue/sched system)

- Estimates of:
  - current number of available nodes (and which ones)
  - average (and variance) of wait times
- Ability to query current (on-line) performance of an application
  - an interface to get job and system information at intermediate point in the job
  - right now this is only available at the end of a run

Information

- Want some information that needs very frequent updates
- knowing the quality of the information is also important
  - it's age (timestamp?), and how accurate it is